Claims

What is claimed is:

1. A system for providing an application component where the application component enables a

5   service to be managed as an independent entity comprising:

a context for containing logic and data associated with a service session;

a façade for containing context-independent

10   service logic wherein the façade is not associated with the service session; and

an event portal for providing entry and exit interfaces.

2. The system of claim 1 further comprising

15   management logic for defining operations, administration and management behavior.

3. The system of claim 1 further comprising management logic for defining appearance of the application component.

20   4. The system of claim 1 further comprising a wiring tool to configure a connection between the event portal of the application component to another event portal of a second application component.

5.   The system of claim 4 wherein the wiring tool connects one or more outgoing events from the event portal to one or more incoming events of an event portal associated with the entity.

6.   The system of claim 4 wherein the wiring tool provides the ability to create service variants by modifying connections between application components.

7.   The system of claim 4 wherein the connection does not require hardcoding thereby enhancing flexibility in changing connections.

8.   The system of claim 4 wherein wiring definitions are uploaded to a service execution engine wherein the service execution engine creates the connection.

9.   The system of claim 1 wherein the application component is network independent.

10.   The system of claim 1 wherein the application component encapsulates protocol specific interactions and presents a homogenous interface to other components.

11. The system of claim 1 wherein the application component is network independent and protocol independent.

12. The system of claim 4 wherein the connection is postponed until after the application component is created.

13. The system of claim 1 wherein the service comprises more than one application component where application components are developed by separate developers thereby enabling parallel development.

14. The system of claim 4 wherein runtime context event subscription is established dynamically based on static event subscription definition.

15. The system of claim 4 wherein contexts of different application components pertaining to a service session are maintained in a context envelope.

16. The system of claim 15 wherein the contexts are added dynamically to the context envelope as the contexts are invoked by service logic.

17. The system of claim 1 wherein one or more service variants are selected by the façade for each service session.

18. The system of claim 17 wherein the application component contains a single template which executes by default.

19. The system of claim 1 wherein the application component incorporates one or more of data storage schemas, variables, constants and configuration items.

20. The system of claim 19 wherein the one or more of data storage schemas, variables, constants and configuration items are exported from the application component.

21. The system of claim 19 wherein the one or more of data storage schemas, variables, constants and configuration items are imported by the application component.

22. The system of claim 20 wherein a wiring tool connects an exported item from the application component with an imported item in another application component.

23. The system of claim 1 wherein one or more protocol specific interactions are encapsulated to present a homogenous interface to other one or more application components.

24. A method for providing an application component where the application component enables a service to be managed as an independent entity comprising the steps of:

5   maintaining logic and data associated with the service in a context;

maintaining context-independent service logic in a façade wherein the façade is not associated with the service; and

10   providing entry and exit interfaces.

25. The method of claim 24 further comprising the step of enabling management logic to define operations, administration and management behavior.

26. The method of claim 24 further comprising

15   the step of enabling management logic to defining appearance of the application component.

27. The method of claim 24 further comprising the step of providing a wiring tool to configure a connection between the event portal of the

20   application component to another event portal of a second application component.

28. The method of claim 27 wherein the wiring tool connects one or more outgoing events from the

event portal to one or more incoming events of an event portal associated with the entity.

29. The method of claim 27 wherein the wiring tool provides the ability to create service variants by modifying connections between application components.

30. The method of claim 27 wherein the connection does not require hardcoding thereby enhancing flexibility in changing connections.

31. The method of claim 27 wherein wiring definitions are uploaded to a service execution engine wherein the service execution engine creates the connection.

32. The method of claim 24 wherein the application component is network independent.

33. The method of claim 24 wherein the application component encapsulates protocol specific interactions and presents a homogenous interface to other components.

34. The method of claim 24 wherein the application component is network independent and protocol independent.

35. The method of claim 27 wherein the connection is postponed until after the application component is created.

36. The method of claim 24 wherein the service comprises more than one application component where application components are developed by separate developers thereby enabling parallel development.

37. The method of claim 27 wherein runtime context event subscription is established dynamically based on static event subscription definition.

38. The method of claim 27 wherein contexts of different application components pertaining to a service session are maintained in a context envelope.

39. The method of claim 38 wherein the contexts are added dynamically to the context envelope as the contexts are invoked by service logic.

40. The method of claim 24 wherein one or more service variants are selected by the façade for each service session.

41. The method of claim 40 wherein the application component contains a single template which executes by default.

42. The method of claim 24 wherein the application component incorporates one or more of data storage schemas, variables, constants and configuration items.

5   43. The method of claim 42 wherein the one or more of data storage schemas, variables, constants and configuration items are exported from the application component.

44. The method of claim 42 wherein the one or
10 more of data storage schemas, variables, constants and configuration items are imported by the application component.

45. The method of claim 42 wherein a wiring tool connects an exported item from the application
15 component with an imported item in another application component.

46. The method of claim 24 wherein one or more protocol specific interactions are encapsulated to present a homogenous interface to other one or more
20 application components.